

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): E. Sadot et al.
Case: 501022-A-01-US (Sadot)
Serial No.: 09/915,609
Filing Date: July 26, 2001
Group: 2143
Examiner: Kyung H. Shin

Title: Secret Session Supporting Load Balancer

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants (hereinafter "Appellants") hereby appeal the final rejection dated April 18, 2006 of claims 1-22 of the above-identified application.

REAL PARTY IN INTEREST

The present application is currently assigned to Avaya Inc. or a subsidiary thereof. Avaya Inc. is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences.

STATUS OF CLAIMS

The present application was filed on July 26, 2001 with claims 1-22. Claims 1-22 remain pending in the application. Claims 1 and 19 are the independent claims. Each of claims 1-22 stands rejected under 35 U.S.C. §103(a). Claims 1-22 are appealed.

STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method of load balancing messages to servers of a server farm. The method includes the step of configuring the load balancer with information specifying a pre-assignment of different groups of session ID values to respective ones of the servers, with each of the servers being operative to assign session ID values from its associated one of the pre-assigned groups to sessions handled by that server. The method further includes the steps of the load balancer determining, for at least some client messages including a non-empty session ID field, which server or sub-group of servers is associated with the ID in the ID field, responsive to the configured information, and the load balancer selecting a server to receive each of the at least some client messages, at least partially responsive to the determination.

In an illustrative embodiment, shown in FIG. 1 of the drawings, a server farm 100 comprises servers 102 which respond to requests from clients 106. In establishing secure communication between a given client 106 and a given server 102 using the secure sockets layer (SSL) protocol, the server assigns a session ID to a negotiated session. A load balancer 104 includes a table 130 which lists ranges of SSL session IDs assigned to respective ones of the servers 102. Each of the servers 102 includes a record 132 which identifies the particular range of SSL session IDs that may be assigned by that server to SSL sessions. See the specification at, for example, page 6, lines 1-14. By pre-assigning groups of session ID values to particular servers, the load balancer is more easily able to determine which client messages should be sent to which servers. This advantageously avoids the excessive storage requirements and negative performance impacts that are associated with

conventional load balancers that simply store session IDs with individual server identifiers, without placing any limit on which session IDs can be assigned by which servers. See the specification at page 2, lines 13-32, and page 3, lines 2-6.

Independent claim 19 is directed to a load balancer comprising a memory unit, an input interface and a load balancing unit. The memory unit is adapted to store configured information specifying a pre-assignment of different groups of session ID values to respective servers, each of the servers being operative to assign session ID values from its associated one of the pre-assigned groups to sessions handled by that server. An illustrative embodiment is the load balancer 104 shown in FIG. 1. See the specification at, for example, page 5, lines 22, to page 7, line 2.

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-5, 7-12 and 14-22 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,772,333 (hereinafter “Brendel”) in view of U.S. Patent No. 5,774,668 (hereinafter “Choquier”) and in further view of U.S. Patent No. 6,138,120 (hereinafter “Gongwer”).

2. Claims 6 and 13 are rejected under 35 U.S.C. §103(a) as being unpatentable over Brendel, Choquier and Gongwer in view of U.S. Patent No. 6,611,498 (hereinafter “Baker”).

ARGUMENT

1. §103(a) Rejection of Claims 1-5, 7-12 and 14-22

Claims 1, 4, 7-10 and 14-22

A proper *prima facie* case of obviousness requires that the cited references when combined must “teach or suggest all the claim limitations,” and that there be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to combine the references or to modify the reference teachings. See Manual of Patent Examining Procedure (MPEP), Eighth Edition, August 2001, §706.02(j).

Appellants submit that the Examiner has failed to establish a proper *prima facie* case of obviousness in the present §103(a) rejection of claims 1 and 19, in that the Brendel, Choquier and Gongwer references, even if assumed to be combinable, fail to teach or suggest all the claim

limitations, and in that no cogent motivation has been identified for combining the references or for modifying the reference teachings to reach the claimed invention. Further, even if it is assumed that a proper *prima facie* case has been established, there are particular teachings in one or more of the references which controvert the obviousness argument put forth by the Examiner.

As indicated previously herein, independent claim 1 is directed to a method of load balancing messages to servers of a server farm. The method includes the step of configuring the load balancer with information specifying a pre-assignment of different groups of session ID values to respective ones of the servers, with each of the servers being operative to assign session ID values from its associated one of the pre-assigned groups to sessions handled by that server. The method further includes the steps of the load balancer determining, for at least some client messages including a non-empty session ID field, which server or sub-group of servers is associated with the ID in the ID field, responsive to the configured information, and the load balancer selecting a server to receive each of the at least some client messages, at least partially responsive to the determination.

The recited method of claim 1 solves a serious problem of the prior art. For example, in establishing secure communication between a client and a server using the SSL protocol, the server assigns a session ID to a negotiated session. This session ID is stored by the server for a predetermined time after the termination of the session. A client can establish an SSL connection based on a previously-negotiated session for which the session ID has been stored by the server, by transmitting an SSL “client hello” message to the server along with the session ID of the previously-negotiated session. This ability to establish SSL connections using previously-negotiated session IDs is referred to as SSL persistency. See the specification at page 1, line 19, to page 2, line 8. The problem arises because a conventional load balancer, in order to ensure SSL persistency, must forward any “client hello” messages from a single client to the same server that negotiated the session ID. The way that the conventional load balancer does this is by storing lists of session IDs with respective server identifiers. This is problematic because of the large amount of storage space needed to store the session IDs with the respective server identifiers. Also, searching and managing large lists of session IDs negatively impacts the performance of the load balancer. See the specification at page 2, lines 13-32.

It is important to note that the Brendel reference cited by the Examiner suffers from the very same problem described above, and therefore teaches directly away from the claimed invention. That is, Brendel teaches an arrangement in which a load balancer must store session IDs with respective server identifiers. This is apparent from, for example, the abstract of Brendel, which states that a load balancer “stores the SSL session ID along with a server assignment that identifies the server that generated the SSL session ID” and for subsequent requests from the same client “uses the SSL session ID to send the requests to the same server.” Thus, a load balancer configured in accordance with the techniques described in Brendel will suffer from the excessive storage space problem and its related negative performance impact as described in the specification at page 2, lines 13-32.

The arrangement recited in claim 1 overcomes this serious disadvantage of Brendel and other conventional load balancers by configuring the load balancer with information specifying a pre-assignment of different groups of session ID values to respective ones of the servers, with each of the servers being operative to assign session ID values from its associated one of the pre-assigned groups to sessions handled by that server. Thus, a given server can only assign session IDs from its assigned group of session ID values. With reference to the illustrative embodiment of FIG. 1, load balancer 104 includes a table 130 which lists ranges of SSL session IDs assigned to respective ones of the servers 102. Each of the servers 102 includes a record 132 which identifies the particular range of SSL session IDs that may be assigned by the server to SSL sessions. See the specification at, for example, page 6, lines 1-14. By pre-assigning groups of session ID values to particular servers, the load balancer is more easily able to determine which client messages should be sent to which servers. This advantageously avoids the excessive storage requirements and negative performance impacts that are associated with Brendel and other conventional load balancers that simply store session IDs with individual server identifiers, without placing any limit on which session IDs can be assigned by which servers.

The Examiner acknowledges that Brendel fails to meet the limitations of claim 1, but argues that the Choquier and Gongwer references supply the missing teachings. More specifically, the Examiner argues that Choquier and Gongwer collectively disclose the limitation of claim 1 which

calls for configuring the load balancer with information specifying a pre-assignment of different groups of session ID values to respective ones of the servers, with each of the servers being operative to assign session ID values from its associated one of the pre-assigned groups to sessions handled by that server. We respectfully disagree with the assertion of the Examiner. In arguing that the above limitation is met by Choquier and Gongwer, the Examiner relies on the teachings in column 15, lines 28-41 of Choquier and column 2, lines 2-5, column 9, lines 52-54, and column 12, lines 54-57 and 62-65 of Gongwer. See the final Office Action at pages 4-5.

The relied-upon teachings from Choquier relate to use of “a randomization technique in selecting a server.” See column 15, lines 22-23. Such random server selection is not only directly contrary to the claimed invention, but clearly inapplicable to the Brendel approach in which the load balancer stores for each SSL session ID a server assignment that identifies the server that generated the SSL session ID. In other words, because Brendel must direct subsequent requests from a given client to the server that previously generated a session ID for that client, Brendel cannot utilize random server selection of the type described in Choquier. To do so would make the Brendel approach unworkable, which is strong evidence that those skilled in the art would not be motivated to make the proposed combination.

The relied-upon teachings from Gongwer fail to supplement the fundamental deficiencies of Brendel and Choquier as identified above. For example, Gongwer in column 9, lines 52-54, refers to assignment of “a new session handle from the pool of unassigned session handles.” However, this is nothing more than what is done in conventional practice as described in the background portion of Appellants’ specification at page 1, lines 29-30. What is not taught by the collective teachings of Gongwer, Choquier and Brendel is the limitation at issue in which a load balancer is configured with information specifying a pre-assignment of different groups of session ID values to respective servers, with each of the servers being operative to assign session ID values from its associated one of the pre-assigned groups to sessions handled by that server.

Accordingly, the collective teachings of the applied references fail to meet the limitations of claim 1, and there is no motivation for the proposed combination in view of the clear teachings away in both Brendel and Choquier.

Independent claim 19 is believed allowable for reasons similar to those identified above in the context of independent claim 1.

Dependent claims 4, 7-10, 14-18 and 20-22 are believed allowable for at least the reasons identified above with regard to their respective independent claims.

Claim 2

Dependent claim 2 further specifies that configuring the load balancer comprises managing a table which lists for at least one of the servers or sub-groups of servers a range of values from which the server may assign session IDs. The Examiner argues that these limitations are shown in column 15, lines 28-41 of Choquier. See the final Office Action, at page 5. However, the relied-upon portion of Choquier relates to random selection of servers, and fails to teach or suggest assignment of a range of values from which a given server may assign session IDs. As indicated above, such random server selection is a direct teaching away the claimed invention.

Accordingly, it is believed that the proposed combination of Brendel, Choquier and Gongwer fails to meet the particular limitations of dependent claim 2.

Claim 3

Dependent claim 3 further specifies that configuring the load balancer comprises managing a table which lists for at least one of the servers or sub-groups of servers, one or more values of a sub-set of the bits of session IDs associated with the server. The Examiner argues that these limitations are shown in Brendel at column 9, lines 7-10 and column 15, line 4. See the final Office Action, at page 6. However, the relied-upon portions of Brendel relates to conventional storage of information indicating which server assigned a particular session ID. There is no teaching or suggestion regarding, for example, listing for a particular server one or more values of a sub-set of the bits of multiple session IDs associated with that server. As previously noted, the relied-upon teachings in Brendel suffer from the very same problems that Appellants have addressed and solved with their invention.

Accordingly, it is believed that the proposed combination of Brendel, Choquier and Gongwer fails to meet the particular limitations of dependent claim 3.

Claim 5

Dependent claim 5 further specifies configuring at least one of the servers with a rule on the session ID values it may assign to sessions. The Examiner argues that these limitations are shown in column 8, lines 35-42 of Brendel. See the final Office Action, at page 6. Appellants respectfully disagree. This portion of Brendel simply provides a mechanism for indicating which server assigned a particular session ID. It does not teach or suggest the placement of any limits whatsoever on the particular session IDs that can be assigned by a given server.

Accordingly, it is believed that the proposed combination of Brendel, Choquier and Gongwer fails to meet the particular limitations of dependent claim 5.

Claim 11

Dependent claim 11 further specifies configuring substantially all the servers by assigning substantially a same number of session IDs to each of the servers. The Examiner apparently relies on the random server selection approach described in Choquier at column 15, lines 28-41. See the final Office Action, at page 9. However, this type of random server selection is directly contrary to the teachings of the claimed invention, and does not involve assignment of particular session IDs to each of a plurality of servers.

Accordingly, it is believed that the proposed combination of Brendel, Choquier and Gongwer fails to meet the particular limitations of dependent claim 11.

Claim 12

Dependent claim 12 further specifies configuring substantially all the servers by assigning different numbers of session IDs to at least two of the servers. The Examiner again relies on the random server selection approach described in Choquier at column 15, lines 28-41. See the final Office Action, at page 9. However, as noted above, this type of random server selection is directly

contrary to the teachings of the claimed invention, and does not involve assignment of particular session IDs to each of a plurality of servers.

Accordingly, it is believed that the proposed combination of Brendel, Choquier and Gongwer fails to meet the particular limitations of dependent claim 12.

2. §103(a) Rejection of Claims 6 and 13

Dependent claims 6 and 13 are believed allowable for at least the reasons identified above with regard to claim 1. The Baker reference fails to supplement the fundamental deficiencies of the Brendel, Choquier and Gongwer references as applied to claim 1.

In view of the above, Appellants believe that claims 1-22 are in condition for allowance, and respectfully request the withdrawal of the §103(a) rejections.

Respectfully submitted,

A handwritten signature in black ink that reads "Joseph B. Ryan". The signature is fluid and cursive, with the first letters of the first and last names being capitalized and prominent.

Date: September 20, 2006

Joseph B. Ryan
Attorney for Appellant(s)
Reg. No. 37,922
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-7517

CLAIMS APPENDIX

1. A method of load balancing messages to servers of a server farm, by a load balancer, comprising:

configuring the load balancer with information specifying a pre-assignment of different groups of session ID values to respective ones of the servers, each of said servers being operative to assign session ID values from its associated one of the pre-assigned groups to sessions handled by that server;

determining, by the load balancer, for at least some client messages including a non-empty session ID field, which server or sub-group of servers is associated with the ID in the ID field, responsive to the configured information; and

selecting, by the load balancer, a server to receive each of the at least some client messages, at least partially responsive to the determination.

2. A method according to claim 1, wherein configuring the load balancer comprises managing a table which lists for at least one of the servers or sub-groups of servers a range of values from which the server may assign session IDs.

3. A method according to claim 1, wherein configuring the load balancer comprises managing a table which lists for at least one of the servers or sub-groups of servers, one or more values of a sub-set of the bits of session IDs associated with the server.

4. A method according to claim 1, wherein configuring the load balancer comprises providing a function which correlates between session IDs and the server which assigned the session ID.

5. A method according to claim 1, comprising configuring at least one of the servers with a rule on the session ID values it may assign to sessions.

6. A method according to claim 5, wherein configuring the load balancer comprises configuring through a user interface, which configures both the load balancer and at least one of the servers responsive to user instructions.

7. A method according to claim 5, wherein configuring the load balancer comprises configuring automatically by a module running on the load balancer, which transmits configuration instructions to at least one of the servers.

8. A method according to claim 7, wherein configuring automatically by the load balancer comprises configuring responsive to input received from the at least one of the servers.

9. A method according to claim 5, wherein configuring at least one of the servers comprises configuring substantially all the servers in the farm with respective sub-groups of allowed session IDs which do not include common session IDs.

10. A method according to claim 9, wherein at least some of a plurality of available session IDs are not assigned to any of the servers.

11. A method according to claim 9, wherein configuring substantially all the servers comprises assigning substantially a same number of session IDs to each of the servers.

12. A method according to claim 9, wherein configuring substantially all the servers comprises assigning different numbers of session IDs to at least two of the servers.

13. A method according to claim 1, wherein configuring the load balancer comprises configuring by a system manager.

14. A method according to claim 1, wherein selecting a server to receive a client message comprises selecting a server which assigned the session ID of the message.

15. A method according to claim 1, wherein selecting a server to receive a client message comprises selecting a server in a sub-group of servers which shares information with a server which assigned the session ID of the message.

16. A method according to claim 1, wherein the client messages comprise SSL client messages.

17. A method according to claim 1, wherein the session ID values comprise application layer ID values.

18. A method according to claim 1, additionally comprising managing a list of ID values actually assigned by one or more servers and determining, by the load balancer, for at least some client messages including a non-empty session ID field, which server or sub-group of servers is associated with the ID in the ID field, responsive to the managed list.

19. A load balancer, comprising:

a memory unit adapted to store configured information specifying a pre-assignment of different groups of session ID values to respective ones of the servers, each of said servers being operative to assign session ID values from its associated one of the pre-assigned groups to sessions handled by that server;

an input interface adapted to receive client messages; and

a load balancing unit which is adapted to select a server to receive at least one of the client messages, at least partially responsive to the contents of the memory unit, and to forward the at least one of the client messages to the selected server.

20. A load balancer according to claim 19, comprising a configuration module adapted to store the configured information in the memory unit.

21. A load balancer according to claim 20, wherein the configuration module is adapted to generate instructions directed to one or more servers on the session ID values they may use.

22. A load balancer according to claim 19, wherein the load balancing unit comprises a comparator adapted to compare at least a portion of at least one of the fields of received client messages to information stored in the memory unit.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None